

In the claims:

Please amend claims 1, 8 and 12-14, in addition to the changes made in an Examiner's Amendment mailed with the Notice of Allowance dated April 28, 2008. The changes in the Examiner's Amendment are assumed to have been entered. The status of the claims is as follows:

1. (Currently Amended) A parallel process execution method with which a plurality of processors execute a plurality of parallel processes produced from a parallel program together with other processes in a time-shared fashion, the method comprising the steps of:

(a) setting a time allocation ratio that determines how much of a given cycle period should be allocated for execution of the parallel program, wherein said setting step sets the time allocation ratio to the parallel program by dividing the given cycle period into a plurality of time slots and determining which process to execute in each time slot of the different processors;

(b) providing a set of criteria ~~before-hand~~ beforehand for use in determining what to execute in a free time slot that has no process assigned, wherein the set of criteria including a throughput-first policy which allows batch processes to run in the free time slot and a turnaround-first policy which allows no batch process to run in the free time slot;

(c) according to the set of criteria, selecting either the throughput-first policy or the turnaround-first policy at each of the processors which process to be executed in the free time slot;

(d) assigning each parallel process of the parallel program to one of the plurality of processors, and starting execution of the assigned parallel processes simultaneously on the plurality of processors; and

(e) stopping the execution of the assigned parallel processes simultaneously on the plurality of processors, when the time elapsed since the start of the parallel processes has reached a point that corresponds to the time allocation ratio that has been set to the parallel program.

2. (Cancelled)

3. (Previously Presented) The parallel process execution method according to claim 1, wherein the processes to be executed in the time slots include interactive processes.

4. (Cancelled)

5. (Cancelled)

6. (Previously Presented) The parallel process execution method according to claim 1, wherein the throughput-first policy gives successively lower priorities to interactive processes, non-parallel processes for execution on a single processor, and parallel processes.

7. (Previously Presented) The parallel process execution method according to claim 1, wherein the processors are grouped into a plurality of nodes, and the method further comprises the steps of:

causing simultaneous interrupts to the nodes;

sending the received interrupts simultaneously to every processor in the nodes; and

causing the processors to start the cycle period in phase with the interrupts.

8. (Currently Amended) A multiprocessor computer which employs a plurality of processors to execute a plurality of parallel processes produced from a parallel program together with other processes in a time-shared fashion, comprising:

a time allocation ratio setting unit that sets a time allocation ratio that determines how much of a given cycle period should be allocated for execution of the parallel program, and provides a set of criteria ~~before hand~~ beforehand for use in determining what to execute in a free time slot that has no process assigned, wherein the set of criteria including a throughput-first policy which allows batch processes to run in the free time slot, and a turnaround-first policy which allows no batch process to run in the free time slot, wherein said time allocation ~~ratio~~ ratio setting unit divides the given cycle period into a plurality of time slots and determines which process to execute in each time slot of the different processors; and

a process execution unit that causes each of the processors to select either the throughput-first policy or the turnaround-first policy, according to the set of criteria, which process to execute in the free time slot, assigns each parallel process of the parallel

program to one of the plurality of processors, starts execution of the assigned parallel processes simultaneously on the plurality of processors, and stops the execution of the assigned parallel processes simultaneously on the plurality of processors when the time elapsed since the start of the parallel processes has reached a point that corresponds to the time allocation ratio.

9. (Cancelled)

10. (Cancelled)

11. (Cancelled)

12. (Currently Amended) A multiprocessor computer system for executing a plurality of parallel processes produced from a parallel program together with other processes in a time-shared fashion, the multiprocessor computer system comprising:

a plurality of nodes, each comprising at least one processor; and

a coordination controller that generates interrupt notifications ~~to a~~ to the plurality of nodes simultaneously;

wherein each node comprises:

a time allocation ratio setting unit that sets a time allocation ratio that determines how much of a given cycle period should be allocated for execution of the parallel program, wherein said time allocation ratio setting unit divides the given cycle

period into a plurality of time slots and determines which process to execute in each time slot of the different processors;

provides a set of criteria ~~before hand~~ beforehand for use in determining what to execute in a free time slot that has no process assigned, wherein the set of criteria including a throughput-first policy which allows batch processes to run in the free time slot and a turnaround-first policy which allows no batch process to run in the free time slot;

a process execution unit that causes each of the processors to select either the throughput-first policy or the turnaround-first policy, according to the set of criteria, which process to be executed in the free time slot; and

a process execution unit assigns each parallel process of the parallel program to the processors, starts execution of the assigned parallel processes simultaneously on the processors, and stops the execution of the assigned parallel processes simultaneously on the processors when the time elapsed since the start of the parallel processes has reached a point that corresponds to the time allocation ratio that has been set to the parallel program.

13. (Currently Amended) A parallel process execution program for use with a plurality of processors to execute a plurality of parallel processes produced from a parallel program together with other processes in a time-shared fashion, the program causing a computer to perform the steps of:

setting a time allocation ratio that determines how much of a given cycle period should be allocated for execution of the parallel program, wherein said setting step

sets the time allocation ratio to the parallel program by dividing the given cycle period into a plurality of time slots and determining which process to execute in each time slot of the different processors;

providing a set of criteria ~~before hand~~ beforehand for use in determining what to execute in a free time slot that has no process assigned, wherein the set of criteria including a throughput-first policy which allows batch processes to run in the free time slot, and a turnaround-first policy which allows no batch process to run in the free time slot;

according to the set of criteria, selecting either the throughput-first policy of the turnaround-first policy at each of the processors which process to be executed in the free time slot;

assigning each parallel process of the parallel program to one of the plurality of processors, and starting execution of the assigned parallel processes simultaneously on the plurality of processors; and

stopping the execution of the assigned parallel processes simultaneously on the plurality of processors, when the time elapsed since the start of the parallel processes has reached a point that corresponds to the time allocation ratio that has been set to the parallel program to allocate the given cycle period.

14. (Currently Amended) A computer-readable storage medium storing a program for use with a plurality of processors to execute a plurality of parallel processes produced from a parallel program together with other processes in a time-shared fashion,

the program when ~~execute~~ executed by a plurality of processors causing the computers to perform the steps of:

setting a time allocation ratio that determines how much of a given cycle period should be allocated for execution of the parallel program, wherein said setting step sets the time allocation ratio to the parallel program by dividing the given cycle period into a plurality of time slots and determining which process to execute in each time slot of the different processors;

providing a set of criteria ~~before hand~~ beforehand for use in determining what to execute in a free time slot that has no process assigned, wherein the set of criteria including a throughput-first policy which allows batch processes to run in the free time slot, and a turnaround-first policy which allows no batch process to run in the free time slot;

according to the set of criteria, selecting either the throughput-first policy or the turnaround-first policy at each of the processors which process to be executed in the free time slot;

assigning each parallel process of the parallel program to one of the plurality of processors, and starting execution of the assigned parallel processes simultaneously on the plurality of processors; and

stopping the execution of the assigned parallel processes simultaneously on the plurality of processors, when the time elapsed since the start of the parallel processes has reached a point that corresponds to the time allocation ratio that has been set to the parallel program to allocate the given cycle period.